基于 TextRank 的自动摘要优化算法 *

李娜娜 [1,2]，刘培玉 [1,2†]，刘文锋 [1,3]，刘伟童 [1,2]

(1. 山东师范大学 信息科学与工程学院，济南 250358; 2. 山东省分布式计算机软件新技术重点实验室，济南 250358; 3. 菏泽学院 计算机学院，山东 菏泽 274015)

摘　要：在对中文文本进行摘要提取时，传统的 TextRank 算法只考虑节点间的相似性，忽略了文本的其他重要信息。首先，针对中文单文档，在现有研究的基础上，使用 TextRank 算法，一方面考虑句子间的相似性，另一方面，使 TextRank 算法与文本的整体结构信息、句子的上下文信息等相结合，如文档句子或者段落的物理位置、特征句子、核心句子等有可能提升权重的句子，来生成文本的摘要候选句群；然后对得到的摘要候选句群做冗余处理，以除去候选句群中相似度较高的句子，得到最终的文本摘要。最后通过实验验证，该算法能够提高生成摘要的准确性，表明了该算法的有效性。

关键词：摘要提取；TextRank；结构信息；候选摘要句群；冗余处理

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.11.0786

Automatic digest optimization algorithm based on TextRank

Li Nana[1,2], Liu Peiyu[1,2†], Liu Wenfeng[1,3], Liu Weitong[1,2]

(1. *School of Information Science & Engineering Shandong Normal University*, *Jinan* 250358, *China*; 2. *Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology*, *Jinan* 250358, *China*; 3. *School of Computer Science*, *Heze University*, *Heze Shandong* 274015, *China*)

Abstract: When abstracting Chinese texts, the traditional TextRank algorithm only considers the similarity between nodes and neglects other important information of the text. Firstly, aiming at Chinese single document, on the basis of existing research, this paper uses TextRank algorithm, on the one hand, it considers the similarities between sentences, on the other hand, TextRank is combined with the overall structural information of texts and the contextual information of sentences, such as the physical position of the document sentences or paragraph, feature sentences, core sentences and other sentences that may increase the weight of the sentence, all are used to generate the digest candidate sentence group of the text. And then, removing high-similarity sentences by redundancy processing technology on the digest candidate sentence group. Finally, the experimental verification shows that the algorithm can improve the accuracy of the generated digest, indicating the effectiveness of the algorithm.

Key words: abstract extraction; TextRank; structure information; the digest candidate sentence group; redundancy processing

0　引言

在自然语言处理（natural language processing）领域中，文本自动摘要的提取是一项比较复杂但意义重大的工作。所谓生成文本摘要就是指利用计算机自动地从原始文献中提取重要句子组成文章摘要的过程。摘要是全面准确地反映某一文献中心内容的简单连贯的短文。文本自动摘要生成的过程中存在一定的困难性，主要表现为：a)计算机并不是人类的大脑，它不能像人类一样在阅读一篇文章后，理解其意义并产生自己的认知，它只能通过统计、计算、机器学习等方法对文档进行机械性的处理，然后从文档中抽取一些能够表达文章主旨的句子，组成文章摘要；b)文本摘要都是通过阅读文章产生的理解，因此必然要了解文章的文本大意，但是计算机并不能够理解文章的含义，也得不到完全符合人类心意的文本摘要。目前的文本自动摘要研究大多倾向于从原始文本中提取能表达文本核心意思的句子，使其尽可能的包含文章所要表达的信息。但无论是提取

收稿日期：2017-11-24；**修回日期：**2018-01-10　　**基金项目：**国家自然科学基金资助项目（61373148）；国家青年自然科学基金资助项目（61502151）；山东省社科规划项目（17CHLJ18，17CHLJ33，17CHLJ30）；山东省自然科学基金资助项目（ZR2014FL010）；山东省教育厅基金资助项目（J15LN34）

作者简介：李娜娜（1991-），女，山东济宁人，硕士研究生，主要研究方向为文本摘要提取；刘培玉（1960-），男（通信作者），山东潍坊人，教授，博导.主要研究方向为自然语言处理与网络信息安全(liupy@sdnu.edu.cn)；刘文锋(1978-)，男，讲师，博士研究生，主要研究方向为自然语言处理；刘伟童(1993-)，女，山东潍坊人，硕士研究生，主要研究方向为情感倾向性分析.

文章中的长句子还是短句子，将其罗列组成文章的摘要，都不可能完整的表达文章的主要含义，也不会达到人们对于摘要的要求。与此同时，还要考虑摘要提取所针对的文本是单文档还是多文档，所以想要使用自动摘要算法生成好的摘要还需要长期的探索。

TextRank 算法[1]是 Mihalcea 和 Tarau 于 2004 年在研究自动摘要提取过程所提出来的，主要是借鉴 Google 公司 PageRank 算法的思路，将句子间的相似关系看成是一种推荐或投票关系，由此来构建 TextRank 网络图，并通过迭代计算至收敛来得到句子的权重值[2]。TextRank 算法具有实现简单、无监督、语言弱相关，同时适用于单文本及多文本处理等优点，但由于其受词频影响大，在提取准确性上，与其他算法相比，并没有太大优势，因此需要对 TextRank 算法进行改进。文献[3~5]将 TextRank 应用于信息的检索，其中文献[3,4]根据一定窗口内词项的共现信息构建无权的 TextRank 网络图：以词语为顶点，词语间的共现关系和语法关系为链，通过整个图的拓扑关系计算词语的权重；而文献[5]则进一步利用词项间的共现频率作为边的权重来构建加权网络，使用加权网络替代无权共词网络文档表示方法的

同时，还提出了基于句子窗口的共词网络构建方式，文章突破了传统的词袋模型，更多的体现了词语在文档中的结构信息。文献[6~8]将 TextRank 应用于关键词的提取，其中文献[6]融合句子和单词之间的三种关系：词与词之间、句子与句子之间、词与句子之间等文章结构信息，可以强化句子和关键词之间的关系，但在计算中只考虑了互信息和向量，框架的稳定性方面还需提高；文献[7]基于 TextRank 算法，利用 ATF*PDF 方法计算文档集中的词语权重，抽取权重较大的实词为候选关键词，并根据候选关键词之间的语义相似关系建立 TextRank 模型，递归计算至收敛，生成关键词序列文章主要考虑词频、词性和词语间的语义关系等信息；而文献[8]则通过引入社会化标签 Tag 的方式来调整 TextRank 词项图中边的权重，并用于计算词项的重要度。文献[1]将标题、段落、特殊句子、句子位置和句子长度等信息引入到 TextRank 网络图的构造中。文献[9]结合了词-句关系和基于图的无监督排序模型，提出了冗余消除技术作为论文方法的补充，从而进一步提高自动汇总的质量。文献[10]讨论了计算句子相似度的方法，并将句子位置、线索词和 TextRank 方法相结合的句子权重计算方案，但只考虑了单个的相似度计算方法。文献[11]利用互信息对文本中词语、句子及段落之间的关联程度进行计算，依据关联程度将整个文本划分成包含不同主题的较小单元,并针对每一单元运用优化的句子权重计算方法进行主题句提取，进而生成文本摘要。文献[12]提出一种基于局部主题关键句抽取的中文自动文摘方法，通过层次分割的方法对文档进行主题分割，从各个局部主题单元中抽取一定数量的句子作为文章的文摘句。文献[13]对文章中可能影响文摘句提取质量的若干特征进行分析,设计了一种基于特征信息提取的句子重要度计算方法,并依此来抽取文摘句以生成摘要。文献[14]考虑词的频率、词性、词的位置、词长等因素，构建词语权重计算公式来表达主题的词和短语具有较高的权重；对句子权重的计算融入句子的内容、位置以及线索词的作用和用户偏好等因素；摘要的生成考虑到候选文摘句的相似性,避免了冗余信息的加入。对摘要的评估进行了从句子粒度到词语粒度的改进。文献[15]充分考虑文章结构和上下文信息的融合，对句子进行综合打分，便于更好地识别文本中的重要句子。文献[16]通过构造词的语义距离计算主题句之间的语义距离，消除摘要的冗余度实现摘要的约简，但没有考虑《同义词词林》中的未登录词。

由此可以看出，大部分的研究只考虑了篇章结构或文章中上下文信息、句子信息等片面的因素，并没有综合考虑各方面因素对生成摘要的影响。本文对近年来自动摘要改进算法做了总结，在现有研究的基础上，结合文档的篇章结构信息及句子的上下文信息，对 TextRank 算法进行改进，并对得到的摘要候选句子做冗余处理，使得到的摘要既简练又包含丰富的信息。

1 TextRank

传统的 TextRank 算法是一种基于图的无监督方法，用于为文本生成关键字和摘要。PageRank 算法是一种链接分析算法，被 Google 搜索引擎用于进行网页排序，它是衡量网页重要程度的算法。PageRank 算法的主要思想是计算一个网页链接的数量和质量，从而估计这个网页的重要程度。该思想是基于假设：更重要的网页会从其他网页收到更多的链接。受 PageRank 算法的启发，TextRank 算法的主要思想是将文档划分成若干词或句子等的文本单元，这些文本单元构成节点，节点间的相似度构成边，进而形成文本图，然后采用矩阵迭代收敛的方式对节点进行排序，得到关键词或摘要句。

TextRank 网络图的构造[2]如下：假设 $V=\{V_1, V_2, \ldots, V_n\}$ 是由 n 个元素 V_i $(1\le i\le n)$ 所构成的集合。以 V_i 为节点，节点间是否具有相似关系为边，构成有向的 TextRank 网络图

$G=(V,E,W)$，其中 $E\subseteq V\times V$ 为节点间各个边的非空有限集合。

记为 $E=\{(V_i, V_j) \mid V_i \in V \Lambda V_j \in V \Lambda w_{ij} \in W \Lambda w_{ij} \neq 0\}$；$W=\{w_{ij} \mid 1\le i\le n \Lambda 1\le j\le n\}$ 是网络图边的权重集合，w_{ij} 为节点 V_i 与 0 间边的权重值（即相似度大小），可通过各种相似度计算函数（如欧氏距离、Jaccard 或余弦函数等）计算所得。

根据有向网络图 $G=(V,E,W)$ 可得到节点间的一个

$n\times n$ 的相似度矩阵 $SM_{n\times n}$：

$$SM_{n\times n}=\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \cdots & \cdots & \cdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \quad (1)$$

矩阵 $SM_{n\times n}$ 是对称矩阵，因为节点 V_i 对 V_j 的贡献度和 V_j 对 V_i 的贡献度是一样的，且 $SM_{n\times n}$ 对角线上元素的值均为 1。

根据 G 和 $SM_{n\times n}$，迭代计算各个节点的权重，权重计算公式为

$$WS(V_i) = 1 - d + d \times \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (2)$$

其中：$WS(V_i)$是节点V_i的权重值（称为 PR 值）；d 是阻尼系数(0≤d≤1)，表示图中某一节点跳转到其他任意节点的概率，d 一般设置为 0.85；$In(V_i)$是指向节点V_i的所有节点的集合；$Out(V_i)$是节点V_i所指向的所有节点的集合；$|Out(V_i)|$是集合 $Out(V_i)$中元素的个数。式(2)的左侧表示节点V_i的权重（*WS* 即 *weight_sum*），右侧的求和表示每个相邻的节点对本节点的贡献程度。求和的分子w_{ij}表示两个节点间的相似程度大小，分母为一个加权和，$WS(V_j)$表示上一次迭代后节点V_j的权重值。

在使用 TextRank 算法时要注意两点：

a)计算节点的权重要用到节点自身的权重，来进行迭代计算。设每个节点的权重初始值均为 $1/|V|$，即 $B_0 = (1,\ldots,1)^T$，那么一般经过若干次迭代计算后可收敛：

$$B_0 = SM_{n \times n} \cdot B_{i-1} \quad (3)$$

b)收敛判定。当两次迭代的结果 B_i 和 B_{i-1} 差别非常小并接近于零时停止迭代计算，此时算法结束，最终可得到包含各个节点权重值的向量。收敛阈值设为 0.0001。根据权重值大小排序可得到相应排名。

在自动生成文本摘要时，抽取文摘句需根据每个句子权重值的大小进行排序，抽取重要度最高的 *T* 个句子作为候选文摘句群。根据字数或句子数要求，从候选文摘句群中抽取句子组成文摘。

2 文本网络图构造

对文本的预处理和特征提取过程如图 1 所示。以句子为单位，对文本进行预处理，包括分词、分句、分段以及词性标注，进而得到句子的特征项。其次，对特征项进行去除停用词、去除敏感词、词性过滤等处理，去掉无用词，只保留具有特定词性的词项；为降低特征空间的维数，还要删除低频词；采取同近义词归并、聚类和分类等策略的目的是降低后续计算的复杂性，减少摘要冗余度并提高表达效果。必须注意的是，在文本中虽然有些词语字面表示不同，但含义相同，常见的如“电脑”、“Computer”、“计算机”、“PC 机”，进行词频统计时，需将这类词作为同一个词处理。

用集合进行如下表示：设文本 D 包含 n 个句子，S_i (1≤ i≤ n)是文本 D 中依次出现的句子，文本 D 表示为 D={$S_1, S_2,\ldots, S_n$}。参照文献[2]，经过图 1 的处理可以得到：

a)文本特征词向量，记为

$D_{key} = [key_1 : fre_1,\ldots, key_j : fre_j,\ldots, key_h : fre_h]$ (1≤ j≤ h)，h 是该文本中所有特征词的数量，h=|D_{key}|；fre_j 是特征词 key_j 在文本中的词频。

b) S_i 的一维向量

$S_{ih} = [key_{i1} : wfre_{i1},\ldots, key_{ij} : wfre_{ij},\ldots, key_{ih} : wfre_{ih}]$ (1≤ j≤ h)，$wfre_{ij}$ 是特征词 key_{ij} 的词频。如果特征词 key_j 在句子 S_i 中出现，那么相应的 $wfre_{ij}$ 是特征词 key_j 在该句子中的词频；否则，$wfre_{ij}$ 为 0。

$$M_{n \times h} = \begin{bmatrix} w_{11} & \cdots & w_{1h} \\ \cdots & \cdots & \cdots \\ w_{n1} & \cdots & w_{nh} \end{bmatrix}$$

矩阵 $M_{n \times h}$ 由所有 S_{ih} 构成，其中第 i 行表示 S_{ih}，对应的每一列 w_{ij} 为 S_i 中对应的分量 $key_{ij} : wfre_{ij}$，由于句子长度不一致，所以 $M_{n \times h}$ 是一个高维稀疏矩阵。

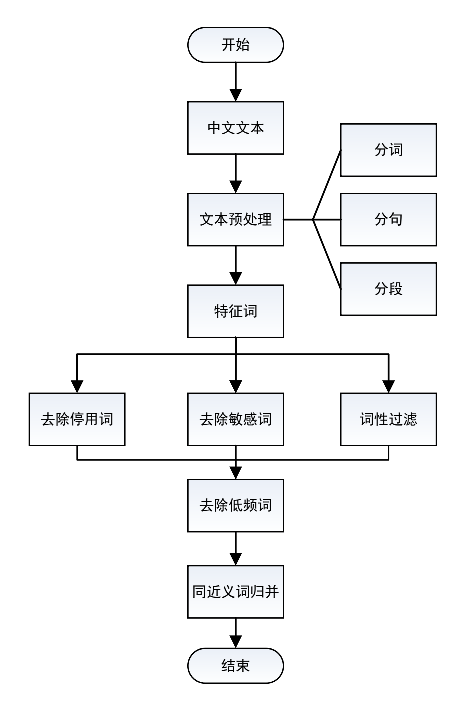


图 1 文本预处理和特征选择流程

本文采用 TF-IDF 方法用于特征词的评估。考虑到文档长度的因素带来的影响，减小词频差异所带来的影响，根据文献[2]，将词频用词频对数来代替。特征词评估函数定义为

$$WS(key_j) = \frac{lg(fre_j + 1) \times lg \frac{N}{N_{key_j}}}{\sqrt{\sum_{j=1}^{h} (lg(fre_j + 1) \times lg \frac{N}{N_{key_j}})^2}} \quad (4)$$

其中：N 为分词工具中词典所包含的特征词的总数，N_{key_j} 为 key_j 在 N 中出现的次数。

将计算结果进行排序，取排名靠前的一定数量的特征词，得到关键词列表。

经过图 1 的处理，得到每个句子的特征词向量。计算各个句子之间的相似度，并作为句子间边的权重。如果两个句子之间没有相似度，即相似度为 0，意味着它们之间不存在相应的边；如果两个句子之间具有一定的相似度，即相似度不等于 0，那么这两个句子之间存在一条边，相似度的数值代表了句子间相似性的大小。

句子间的相似度可以通过欧氏距离、Jaccard、余弦函数或 BM25 等相似度计算函数计算得到。本文采用余弦相似度方法，参照文献[2]，计算公式为

$$w_{ij}=\frac{S_{ih}\cdot S_{jh}}{||S_{ih}||\cdot||S_{jh}||}=\frac{\sum_{1\le m\le h}WS(wfre_{im})\times WS(wfre_{jm})}{\sqrt{\sum_{1\le p\le h}WS(wfre_{ip})^2}\sqrt{\sum_{1\le p\le h}WS(wfre_{jp})^2}} \tag{5}$$

其中：·为向量点积。

矩阵 $SM_{n\times n}$ 由 $M_{\mathrm{n}\times h}$ 与式(5)得到：

$$SM_{n\times n}=M_{n\times h}\bullet M_{n\times h}^{T}=\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \cdots & \cdots & \cdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix} \tag{6}$$

$SM_{n\times n}$ 是句子间的相似度矩阵，权值 w_{ij} 表示句子 S_i 与 S_j 间的相似度。 $SM_{n\times n}$ 为对称阵，其对角线上的元素值均为 1。

构建一个有向的加权 TextRank 网络图的大致过程如下：以文本中各句子为节点，句子间是否具有相似关系为边，句子间的相似度为边的权值，各节点的权重计算公式为

$$WS(S_i)=1-d+d\times\sum_{S_j\in In(S_i)}\frac{w_{ij}}{\sum_{V_k\in Out(S_j)}w_{jk}}WS(S_j) \tag{7}$$

根据第 2 部分，设每个节点的权重初始值均为 1，即 $B_0=(1,\ldots,1)^T$ ，经过若干次的迭代计算后 B_i 可收敛：

$$B_0=SM_{n\times n}\cdot B_{i-1} \tag{8}$$

收敛后的 B_i 包含了各个句子节点的权重值，对权重值的大小进行排序得到相应句子排名。根据一些其他要求，如字数或句子数，并结合他们在文本中的先后顺序，从候选文摘句中抽取排名靠前且符合一定数量的 $N(1\leqslant N\leqslant|D|)$个句子构成文本的摘要。$N$ 的值可通过统计信息来确定。例如：统计分析摘要句子数量与文本句子数量间的比例以确定合适的 N 值。

3　基于 TextRank 的自动摘要优化算法

3.1　文本结构和句子信息

此部分主要叙述本文的改进思路，主要考虑文本的结构以及句子的信息，包括文本结构信息和句子上下文信息。此外，考虑到摘要的新颖性和相关度，对获得的候选摘要句群做冗余优化处理。最后提及摘要的输出问题，目的是保证最终摘要的连贯性和可读性。

3.1.1 句子位置

对于像新闻类等的文章，往往会在第一段交代很多文章主旨信息，在此类文章中，距离文章开始位置越近的段落或者句子应适当提高其权重。根据美国科学家的研究结果：在人工摘要中，选取段首句作为摘要的比例为 85%，选取段尾句作为摘要的比例为 7%。

基于以上信息，可对文档进行分类处理，即普通文档和新闻类文章或者在文章结构上与新闻类文章相似的文档。对于此类文档，距离文章开始位置越近的段落或者句子应适当提升其权重；另一类文档可以根据段落位置和段落中句子的位置进行加权。对首段中越靠前的句子给予越大的权重提升，末段中越靠后的句子给予越小的权重提升。因为收敛后的权重矩阵 B_i 仍然按照句子的先后顺序排序，因此可根据句子的位置调整相应的权重。

设文章首段中有 u 个句子，末段由 v 个句子组成，对 B_i 中句子的权重调整可以通过转移矩阵 $TM_{n\times 1}$ 来实现，$TM_{n\times 1}=[sw_1,\ldots,sw_s,\ldots,sw_{n-r+1},\ldots,sw_n]^T$ 。$TM_{n\times 1}$ 中前 u 个 $sw(1\leq j\leq u)$的值采用依次递减的方式，而后 v 个句子采用依次递增的方式。

$$\begin{cases}(1+e_1)-(j-1)\times\frac{e_1}{u},1\le j\le u\\ 1,s<j<n+1-v\\ 1+(j-n+v)\times\frac{e_2}{v},n+1-v\le j\le n\end{cases}$$

其中： e_1 和 e_2 均为调整阈值，本文中 e_1=0.5， e_2=0.1。

通过矩阵相乘 $B_{\mathrm{i}+1}=TM_{n\times 1}\cdot B_i$ ，实现对最终权重的调整。

3.1.2 文章标题

将文章中标题句子记为 S_0 ，其特征词向量表示为 $S_0=[\mathrm{k}_{01},\ldots,k_{0h'}]^T$ ，h'是扩展后包含标题及句子的特征词的数量。主要考虑的因素为：

a)每个句子与文章标题的相似度。相似度越高，则该句子的权重越高；反之，句子权重越低。

由第 2 部分可知在 TextRank 算法中，经过多次迭代计算后，每个句子的数值趋于稳定，说明句子的稳定值与初始值无关，只与其他句子对本句子所做的贡献度有关。利用标题 S_0 与各个句子 S_i 间的相似度，调整收敛后 B_i 中的句子权重。

调整规则：如果文章的标题与句子的特征词完全相同，即相似度为 1，则将该句子的最终权重放大 2 倍（将权重放大的倍数太高，易造成数据的极性化，造成最终的错误）；其他情况下保持原权重不变。

b)文章中各个句子的特征词是否在标题中也同时出现。如果特征词在标题中出现，则适当提升其词频的权重；否则，保持词频权重不变。

3.1.3 特殊句子

根据中文文章的特点，在一个文章中，如果一个句子自成一段，那么这个段落往往起着“承上启下”或者“过渡句”的作用。文章中还可能存在一些小标题，自成一段。这些具有特殊性的关键句子一般具有高概括性、精炼性的特点，符合摘要本身的要求，所以有更大的可能性成为摘要的一部分。对于文章中带有启发词汇的句子，比如带有“总之”“综上所述”“总而言之”等能够表达总结词汇的句子，是对文章或者段落的总结，需要对此类句子的权重进行适当提升。

对于上述讨论的句子可以给予更大的权重，处理方法类似于前面(1)部分对特殊段落中句子位置信息中对首段和末段句子的权重提升。但是，在本部分中需要对此类句子进行筛选，因为文章中往往也存在一些没有意义的短句子(一般字数小于或等于 6)自成一段，这些句子就没有提升权重的必要。此外，类似于问句等这样不适合作为摘要的句子，它们的句子权重也没有提升的必要。

3.1.4 句子权重

众所周知，在 TextRank 算法中，句子的重要程度是由句子本身所得到的其他句子的“投票”数量和质量决定的，得票越多，句子越重要。当句子权重得到提升后，与该句子相关联的句子权重也应得到相应的提升，这就需要将更新的句子权重进行传递，使最终的计算结果更加准确。将首段句子、末段句子、关键句子、独立存在的句子等进行标记，并将这些句子传送出去的权重放大，使与之关联的句子都获得更加精确的权重值。

在此部分，设置一个阈值 α（α>1）作为界限来确定那些与之关联的句子权重要放大的倍数，使关联性强的句子获得更大的权重值，即放大相似度矩阵 $SM_{n\times n}$ 中第 i 行的值。还要确定一个余弦相似度值用来确定放大哪些关联句子的权重。

3.1.5 过滤句子长度

一个句子能否作为摘要候选句，该句子本身的长度也是一个重要的条件，过长或过短的句子都不应该作为要生成的摘要的候选句。例如，经过预处理后不包含基本特征词的句子可以直接忽略。本文中，将长度系数 C_L >0.8 以及 C_L <0.2 的句子去掉。句子长度系数定义为

$$C_L = \frac{L}{L_m} \tag{9}$$

其中：L 为句子的长度，L_m 为最长句子的长度。

3.1.6 冗余处理

摘要应该具有的硬性评价指标包括新颖性和相关性。新颖性是指候选句子包含的冗余信息少，尽可能的每句话都可以独立的表示出一种独立的意思。相关性是指摘要所用的句子最能够代表该文档的意思。

因此对最终获得的摘要候选句子，为了使生成的摘要尽可能的包含原始文档的信息含量，就要减少具有相同信息的句子重复出现，需要对其进行再一次的冗余处理，所以将相似度较高的句子进行减分或者是去除操作。

本文在选择摘要内容时，利用余弦相似度来判别冗余信息。在此部分，一些研究会在计算句子间相似度时，引入惩罚因子，对所有的句子进行重新打分，公式为：$a\times score(i)+(1-a)\times similartity(i,i-1)$。序号 i 表示排序后的顺序，排序第一的句子不需要重新计算，从第二句开始后面的句子必须和前一句的相似度进行惩罚，也就是 MMR(maximum margin relevence)。当句子间的相似性较大时，按这几个句子在候选文摘句子中出现的顺序进行选择，将剩余的句子作为冗余句子，并从候选摘要句群中删除。

值得注意的一个问题是：对候选摘要句子作冗余处理之后，可能会导致最终生成的摘要句子数不符合要求，当出现此类情况时，需要从根据句子权重值大小排序得到的句子重要性排名中，依次向下取出一定数目的句子，做相似度比较后，将相似度较小的句子加入最终摘要，直到句子数符合最终摘要的句子数目要求。例如：最终摘要句子的数目是 5，在得到的句子重要性排名中，选择排名前 8 的句子作为候选摘要句，对候选摘要句子作冗余处理之后，得到的摘要句子数为 4，那么本文句子重要性排名中，继续向下选择排名第 9 的句子，与得到的 4 句摘要作相似度计算，若相似度小，则将其加入最终摘要；若相似度较大，则继续向下选择第 10 句……，直到找到所规定的 5 句摘要句子。对于此类问题，另一个解决办法是：按一定的比例从句子重要性排名中选择候选摘要句子，经过冗余处理后，按照句子出现的先后顺序，只选符合目标的句子数目即可。

3.1.7 摘要输出

输出结果是经过冗余处理后的形成的摘要。因为每个句子都是从不同的段落中选择出来的，所以要考虑摘要句子的可读性。若硬生地将其连接成摘要，不能保证句子间的衔接和连贯。所以，在本文中，将排序的句子按原文中的顺序输出，可在一定程度下，保证一点连贯性。

3.2 算法实现

本文算法的具体实现如下。

a)对文本进行预处理和特征提取，得到文本和句子相关的特征向量。

b)对文章进行句子长度过滤。

c)计算每个句子与文章标题的相似度，调整词频权重。

d)检查文章中各句子的特征词是否在标题中出现，调整词频权重。

e)综合段落位置、句子位置、特殊句子处理以及相关句子权重的传递，再次调整词频权重。

f)对综合调整后的词频权重矩阵进行迭代计算，直至收敛。

g)根据权重值的大小进行排序，得到相对应的句子，组成摘要候选句群。

h)对摘要候选句群进行句子长度过滤，去除疑问句等不适合做摘要的句子。

i)对候选摘要句群做冗余处理。

j)输出最终摘要。

本文算法的流程图如图 2 所示。

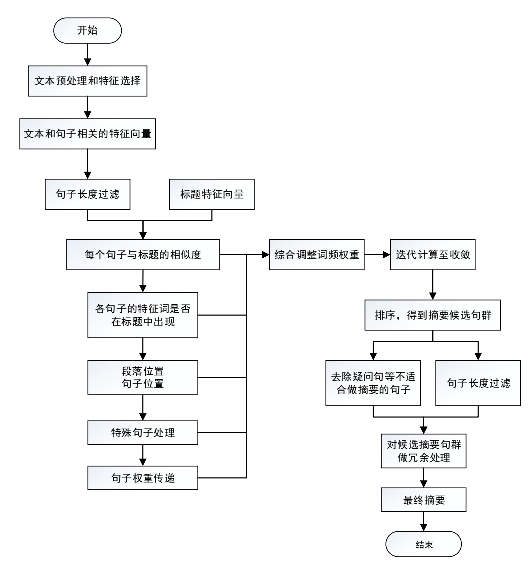


图 2　算法流程图

4 实验

本文设计了两个实验来验证本文方法的有效性：a)将传统 TextRank 算法、现有研究中基于 TextRank 作出改进的算法与本文算法产生的结果进行比较。并计算准确率 P、召回率 R 和平均 F 值；b)将使用本文算法产生的摘要与网上的在线摘要系统生成的摘要进行对比。

4.1 本文算法与传统 TextRank 算法的比较

从各大新闻网站采集若干篇文章，首先对其进行人工标注得到其摘要。然后通过本文算法和传统的 TextRank 算法、参考文献中基于 TextRank 作出相应改进的算法分别生成每篇文章的摘要，通过平均准确率 P、平均召回率 R 和平均 F 值来分析两种方法自动生成摘要与人工标注摘要的差异度。

平均准确率 P、平均召回率 R 和平均 F 值的定义如下：

$$P = \frac{\sum_{i=1}^{n} \frac{|a_i \cap b_i|}{|a_i|}}{n} \quad (10)$$

$$R = \frac{\sum_{i=1}^{n} \frac{|a_i \cap b_i|}{|b_i|}}{n} \quad (11)$$

$$F = \frac{2PR}{P+R} \quad (12)$$

其中: a_i 表示通过使用算法生成的第 i 篇文章的摘要， b_i 表示第 i 篇文章通过人工标注生成的摘要。

实验结果如表 1 所示。

该部分列举 TextRank、iTextRank、TextRankExt 和本文算法的实验结果。其中，TextRank 是传统的基准算法；iTextRank 是文献[2]所提出的方法，该方法将标题、段落、句子位置等信息引入到 TextRank 网络图的构造中，结合了词频统计特征、上下文和语义信息；TextRankExt 是文献[18]中提出的方法，该方法将句法、语义和统计方法考虑其中，共同作用于句子的评分。

通过对表 1 的数据进行对比可以发现，本文的算法产生的摘要在平均准确率 P、平均召回率 R 和平均 F 值方面均比其他 3 个算法生成的摘要效果要好，更加接近于人工标注所产生的摘要，iTextRank 与 TextRankExt 的 P、R、F 值是相近的，说明生成的摘要效果是近似的，但结果均优于 TextRank。实验结果说明将文本的整体结构信息、句子的上下文信息等因素考虑到摘要提取的过程之中，并使其与 TextRank 算法相结合，在多个因素的共同作用下，使形成的文章摘要质量获得提升，并且本文算法对候选摘要句群做了进一步的冗余优化处理，使得生成的摘要比 iTextRank 算法产生的摘要效果更好。

表 1 各种算法的实验结果对比

算法		2 句	3 句	4 句	5 句
TextRank	P	0.369	0.344	0.322	0.325
	R	0.328	0.346	0.352	0.381
	F	0.347	0.345	0.342	0.351
iTextRank	P	0.388	0.334	0.330	0.338
	R	0.389	0.389	0.409	0.424
	F	0.388	0.359	0.365	0.376
TextRankExt	P	0.375	0.326	0.332	0.340
	R	0.393	0.410	0.432	0.455
	F	0.384	0.363	0.375	0.389
本文算法	P	0.453	0.437	0.422	0.413
	R	0.435	0.471	0.493	0.530
	F	0.444	0.453	0.455	0.464

4.2 与在线语义分析系统所生成的自动摘要的对比

将本文算法生成的摘要、在线语义分析系统（http://ictclas.nlpir.org/nlpir/）生成的摘要与人工标注获得的摘要进行比较。在该部分，以展现生成的摘要效果为目的，结合实际需要和客观因素，展现本文算法和在线摘要生成的摘要句数为 2~3 句。结果如表 2 所示。

表 2 摘要结果对比

序号	人工标注摘要	在线语义分析系统	本文算法
1	全球智能机出货量在今年第三季度比以往增长了 2.7%，苹果智能手机出货量依然高于华为。华为曾经有过在出货量上超越苹果的短暂时刻，但利润并没有超越苹果。从未来趋势来看，华为出货量超越苹果可能越来越难，原因如下：利润层面的差距还会继续拉大；iPhoneX 是其最强大的对手，会收割不少客户；华为没有抓住印度市场，小米 OV 在印度市场占据上风；华为近两年战略核心是对中高端市场的定位，更强调以利润优先，导致其主力布局都在国内与欧美市场，但与苹果三星抢夺市场的难度也大过以往。	市场研究机构 IDC 最新的研究报告显示，智能手机市场在今年第二季度出现罕见萎缩后，全球智能手机出货量在今年第三季度达到 3.731 亿部，与 2016 年第三季度的 3.634 亿部相比，增长了 2.7%。因此，从国内竞争对手小米与印度市场机遇丧失，以及在高端市场被 iPhoneX 强势压制的种种不利环境来看，华为出货量要超越苹果越来越难了。	市场研究机构 IDC 最新的研究报告显示，智能手机市场在今年第二季度出现罕见萎缩后，全球智能手机出货量在今年第三季度达到 3.731 亿部，与 2016 年第三季度的 3.634 亿部相比，增长了 2.7%。从未来趋势来看，华为在出货量超越苹果可能越来越难了，利润层面可能差距还会继续拉大。
2	与 iPhoneX 相比，iPhone8 呈现出门庭若市的局面，主要有三个原因：iPhoneX 抢了 iPhone8 的风头、iPhone8 本身的亮点太少以及用户不买 iPhone 心机，根其是否便宜关系不大。	相比较 iPhoneX 的疯狂，iPhone8 可谓是苹果历史上最不显眼的新机了，其锋芒完全被 iPhoneX 给盖住，与 iPhone8 的门可罗雀相比，iPhoneX 则是门庭若市的局面，虽然黄牛党被坑了，但依然挡不住用户对 iPhoneX 的热情。第三，用户买不买 iPhone 新机，跟其是否便宜关系不大。便宜、降价是没用的，愿意买苹果的人，肯定是选择 iPhoneX。	相比较 iPhone X 的疯狂，iPhone8 可谓是苹果历史上最不显眼的新机了，其锋芒完全被 iPhone X 给盖住，与 iPhone 8 的门可罗雀相比，iPhone X 则是门庭若市的局面，虽然黄牛党被坑了，但依然挡不住用户对 iPhone X 的热情。第一，iPhone X 抢了 iPhone8 的风头。第二，iPhone8 本身的亮点太少。

根据表 2，对比人工标注摘要，可以看出：无论是在线摘要系统生成的摘要还是本文算法生成的摘要，二者均能够紧扣文章的关键词，使其表达围绕主题思想，能较好的表达文章的主旨。在展示出的摘要中，不难发现，二者存在部分重叠的句子，其他句子虽然在一定程度上有差距，但是不影响基本的中心思想表达。且本文算法生成的摘要，也具有较好的语意连贯性，便于读者的理解。

5　结束语

自动摘要的生成是自然语言处理领域的研究重点。本文总结了近年来自动摘要的生成算法，并在 TextRank 算法的基础上，结合段落、句子位置、特殊段落和句子等与文本整体结构和句子上下文信息，在得到摘要候选句群后。又做了进一步的冗余处理，删除了相似度较大的摘要候选句，使得

到的摘要更加精炼，将文章要的意思表达的更完整。在实验部分，本文的方法得到了很好的验证。下一步工作是将本文的算法应用到不同类型的文本中，即针对某一类型的文章找到适合的摘要提取方法。

参考文献：

[1] Mihalcea R, Tarau P. TextRank: bringing order into texts [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2004: 404-411.

[2] 余珊珊，苏锦钿，李鹏飞. 基于改进的 TextRank 的自动摘要提取方法 [J]. 计算机科学, 2016, 43 (6): 240-247.

[3] Blanco R, Lioma C. Random walk term weighting for information retrieval [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2007: 829-830.

[4] Blanco R, Lioma C. Graph-based term weighting for information retrieval [J]. Information Retrieval, 2012, 15 (1): 54-92.

[5] 陆伟，程齐凯. 一种基于加权网络和句子窗口方案的信息检索模型 [J]. 情报学报, 2013, 32 (8): 797-804.

[6] Wan X, Yang J, Xiao J. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction [C]// Proc. of the 45th Annual Meeting of the Association of Computational Linguistics. [S. l.] : Association for Computational Linguistics, 2007: 552-559.

[7] 杨洁，季铎，蔡东风，等. 基于 TextRank 的多文档关键词抽取技术 [C]// 第四届全国信息检索与内容安全学术会议论文集（上）. 2008: 397-404.

[8] 李鹏，王斌，石志伟，等. Tag-TextRank: 一种基于 Tag 的网页关键词抽取方法 [J]. 计算机研究与发展, 2012, 49 (11): 2344-2351.

[9] Fang C, Mu D, Deng Z, *et al.* Word-sentence co-ranking for automatic extractive text summarization [J]. Expert Systems with Applications An International Journal, 2017, 72 (C): 189-195.

[10] 曹洋. 基于TextRank算法的单文档自动文摘研究 [D]. 南京：南京大学, 2016.

[11] 刘星含，霍华. 基于互信息的文本自动摘要 [J]. 合肥工业大学学报：自然科学版, 2014, 37 (10): 1198-1203.

[12] 徐超，王萌，何婷婷，等. 基于局部主题关键句抽取的自动文摘方法 [J]. 计算机工程, 2008, 34 (22): 49-51.

[13] 叶星火，胡珀，张小鹏. 基于特征信息提取的中文自动文摘方法 [J]. 计算机应用与软件, 2008, 25 (05): 31-32.

[14] 蒋昌金，彭宏，陈建超，等. 基于主题词权重和句子特征的自动文摘 [J]. 华南理工大学学报：自然科学版, 2010, 38 (07): 50-55.

[15] 胡珀. 融合上下文信息的自动文摘研究 [D]. 武汉：武汉大学, 2013.

[16] 程传鹏，杨要科. 自动文摘中的冗余句消除方法 [J]. 计算机应用, 2011, 31 (12): 3275-3277.

[17] 张璐，曹杰，蒲朝仪，等. 基于词句协同排序的单文档自动摘要算法 [J]. 计算机应用, 2017, 37 (07): 2100-2105.

[18] Barrera A, Verma R. Combining syntax and semantics for automatic extractive single-document summarization [C]// Proc of International Conference on Computational Linguistics and Intelligent Text Processing. [S. l.] : Springer-Verlag, 2012: 366-377.